

Create, Read, Update, and Delete (CRUD) DB Operations

Connection

```
<?php
// Database connection
$servername = "localhost";
$username = "root";
$password = "password"; // change it to your password
$dbname = "studentdb";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully<br><br>";
```

PHP DB Command Execution Pattern

For MySQLi (your first code):

- Make a **SQL statement**
- Call `prepare()` → returns a statement object
- Call `bind_param()` → bind values to `?` placeholders
- Call `execute()` → run the query

```
$sql = "INSERT INTO students (name, age, major) VALUES (?, ?, ?)";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("sis", $name1, $age1, $major1);  
$stmt->execute();
```

CREATE - Adding New Students

```
echo "<h2>CREATE - Adding New Students</h2>";

$name1 = "Alice Johnson";
$age1 = 22;
$major1 = "Computer Science";

$sql = "INSERT INTO students (name, age, major) VALUES (?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("sis", $name1, $age1, $major1);

if ($stmt->execute()) {
    echo "Student '$name1' added successfully<br>";
} else {
    echo "Error: " . $stmt->error . "<br>";
}
$stmt->close();
```

READ - Display all student records

- We use `query` command to get results, and display the information from the `fetch_assoc()` function.

Why use `fetch_assoc()` in PHP MySQLi?

- Retrieves each row from the result set as an associative array.
- Lets you access column data by name (e.g., `$row['name']`).
- Makes result processing clear and readable.

```
echo "<h2>READ - All Students</h2>";

$sql = "SELECT id, name, age, major FROM students";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "Found " . $result->num_rows . " students:<br>";
    while($row = $result->fetch_assoc()) {
        echo "ID: " . $row["id"]. " - Name: " . $row["name"].
            " - Age: " . $row["age"].
            " - Major: " . $row["major"]. "<br>";
    }
} else {
    echo "No students found<br>";
}
```

READ - Get specific student by ID

```
echo "<h2>READ - Specific Student (ID = 1)</h2>";

$student_id = 1;
$sql = "SELECT id, name, age, major FROM students WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $student_id);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    echo "Found student: " . $row["name"] . " (Age: " . $row["age"] . ", Major: " . $row["major"] . ")<br>";
} else {
    echo "No student found with ID $student_id<br>";
}
$stmt->close();
```

Prepared Statement

- `$result = $conn->query($sql);`
Runs a direct SQL query, no prepared statements.
- `$stmt->execute(); $result = $stmt->get_result();`
Executes a prepared statement and gets the result. Safer and prevents SQL injection.
- They both return result objects but come from different usage contexts. Prepared statements are preferred for security and flexibility.

UPDATE - Modify existing student record

```
echo "<h2>UPDATE - Updating Student</h2>";

$update_id = 1;
$new_name = "Alice Johnson Updated";
$new_age = 23;
$new_major = "Computer Engineering";

$sql = "UPDATE students SET name = ?, age = ?, major = ? WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("sisi", $new_name, $new_age, $new_major, $update_id);

if ($stmt->execute()) {
    echo "Student with ID $update_id updated successfully<br>";
} else {
    echo "Error updating student: " . $stmt->error . "<br>";
}
$stmt->close();
```

Show updated record

```
echo "Updated student info:<br>";
$sql = "SELECT id, name, age, major FROM students WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $update_id);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    echo "ID: " . $row["id"].
        " - Name: " . $row["name"].
        " - Age: " . $row["age"].
        " - Major: " . $row["major"]. "<br>";
}

$stmt->close();
```

DELETE - Remove a student record

```
echo "<h2>DELETE - Removing Student</h2>";

$delete_id = 2;
$sql = "DELETE FROM students WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $delete_id);

if ($stmt->execute()) {
    echo "Student with ID $delete_id deleted successfully<br>";
} else {
    echo "Error deleting student: " . $stmt->error . "<br>";
}

$stmt->close();
echo "<br>";
```